Deep globally constrained MRFs for Human Pose Estimation

Ioannis Marras, Petar Palasek and Ioannis Patras
School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom
{i.marras, p.palasek, i.patras}@qmul.ac.uk

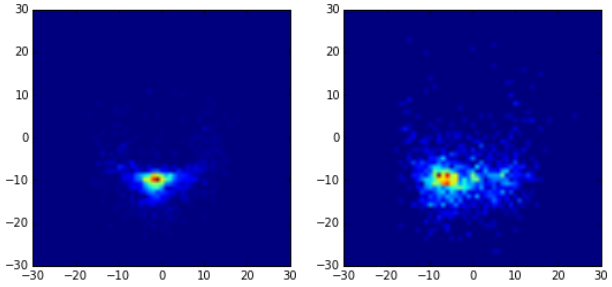
Abstract

This work introduces a novel Convolutional Network architecture (ConvNet) for the task of human pose estimation, that is the localization of body joints in a single static image. We propose a coarse to fine architecture that addresses shortcomings of the baseline architecture in [26] that stem from the fact that large inaccuracies of its coarse ConvNet cannot be corrected by the refinement ConvNet that refines the estimation within small windows of the coarse prediction. We overcome this by introducing a Markov Random Field (MRF)-based spatial model network between the coarse and the refinement model that introduces geometric constraints on the relative locations of the body joints. We propose an architecture in which a) the filters that implement the message passing in the MRF inference are factored in a way that constrains them by a low dimensional pose manifold the projection to which is estimated by a separate branch of the proposed ConvNet and b) the strengths of the pairwise joint constraints are modeled by weights that are jointly estimated by the other parameters of the network. The proposed network is trained in an end-to-end fashion. Experimental results show that the proposed method improves the baseline model and provides state of the art results on very challenging benchmarks.

1. Introduction

The problem of human pose estimation in monocular RGB images, that is the problem of precise localization of important landmarks of the human body, has received substantial attention in the Computer Vision community. Due to the availability of ever larger and more comprehensive datasets [1, 15, 24] and to the success of Deep Learning architectures, especially ConvNets [7, 27, 14, 26], there has been significant progress in this problem over the recent years.

A central issue in human pose estimation, when seen as a special case of a Machine Learning problem with structured outputs, is the enforcement of constraints between the different outputs, that is the enforcement of geometric constraints on the relative locations of the body joints. This is typically modeled at the later layers of a ConvNet. For example, in [26] a MRF that models pairwise relations between different joints is encoded in a single CNN layer. In such a network the filters $e_{a|c}$ encode the conditional probabilities of the location of joint a, given the location of another joint c. A major drawback with such an approach is that a single filter is used to model all of the pairwise relations. This works well when applied to simpler datasets, such as FashionPose, where there is little pose variation and therefore the conditional probabilities have a few distinct modes. However, for more complex datasets the conditionals become more uninformative as they attempt to model pairwise relations under wide variety of poses - for example, the relative location of the head and the hip both in upright and in laying poses. This is evident in Figure 1, where the probability of the hip location given the head is depicted in Figure 1(a) for images of the FashionPose dataset and in Figure 1(b) for images of the MPII benchmark. Other works, such as [5] where the geometric constraints are implicitly modeled in the latest layers of the network, suffer from similar shortcomings. For example, in [5] the last layers that incorporates intensity constraints, imposes pairwise constraints encoded in a single filter (that is, one filter per pair of joints).



(a) (b)

Figure 1. The probability of the hip location given the head location on (a) Fashion Pose and (b) MPII databases.

In this paper, we present a three stage coarse-to-fine Convolutional Network architecture for the task of human pose

estimation. Our model comprises of: a) a coarse ConvNet that provides coarse low(er) resolution heat-maps for the joint locations, b) a part-based constrained MRF model that enforces geometric constraints conditioned on a global projection on a low dimensional manifold, and c) a refinement (coarse to fine) ConvNet, that refines the estimation within windows around the peaks of the coarse heat-maps. The combined model is trained in an end to end fashion to minimize the weighted sum of the costs of each of the three ConvNets. The coarse to fine architecture, that is the coarse and the refinement models, is similar to the baseline model of [25] and is reminiscent of recent works [17] that reuse early layers at the later stages of the architecture. A major challenge in such an architecture is that large inaccuracies of its coarse ConvNet, i.e. when spurious peaks are chosen, cannot be corrected by the refinement ConvNet. For example, the coarse-to-fine ConvNet in [25] relies little on the refinement ConvNet, as evidenced by the low weight assigned to the corresponding cost, resulting only in a moderate improvement of the final localization accuracy.

In this work, we introduce a novel MRF-part based spatial model network between the coarse and the refinement model that enforces spatial geometric constraints between joints (Section 3.2). The proposed MRF model is a general idea that could be applied at other ConvNet systems. It builds on the the geometric model used in [25] that expresses message-passing as convolution operations that can be implemented using ConvNets - the filters expressing the conditional dependencies between the location of different joints. By contrast to it, in our formulation, each of the filters that perform the convolution operations is assumed to be a linear combination of K filters. The weights of this linear combination are the projection of the heat-maps into a K dimensional manifold that encodes global constraints, such as the global pose. Unlike all other architectures, the filters that are applied in our architecture at test time are not static but dynamic, while the projection of the heat-map volume to the low dimensional manifold is performed by a side auto-encoder ConvNet that is jointly trained with the other ConvNets. Thus, the weights are learned by a cost function that combines both a generative term that comes from the auto-encoder ConvNet and a discriminative cost that comes from the heat-map prediction. In this way, the conditionals become more informative as they attempt to model pairwise relations under specific global constraints. Additionally, in our formulation, different pairwise constraints are given different weights. The above constraints amount to a factorization of the filter tensor. Finally, inspired by the work in [6], the message passing procedure is applied in an iterative manner to better mask-out the incorrect joints' activations.

In addition to these central methodological contributions, we make two additional ones that considerably improve the performance. First, we use cropping windows of varying sizes at the peaks of the heat-maps from the coarse network to ensure that the cropped window that is used in the "refinement" network encloses the target joint (Section 3.3). This is in contrast to [25] that uses a fixed window size and therefore relies little on the "refinement" network for the final pose estimation, as evidenced by the fact that a small weight to the cost of the "refinement" network is used during training. Secondly, we use a novel data augmentation and a learning procedure that were both adapted to the difficulty of the specific data instances/images (Section 4.2). More specifically, hard instances (i.e. training images with a large prediction error) were assigned a lower learning rate and were augmented by applying more transformations (rotation, scaling, shearing, stretching and flipping) to them. Furthermore, we have trained our learning framework in a way that is beneficial for our unified learning framework (Section 4.2).

The proposed architecture is trained in an end-to-end fashion. We show experimentally (Section 5) that the combination of the three proposed ConvNets into a unified learning framework: a) significantly outperforms the methods proposed in [25] and [26] and b) provides state of the art results on very challenging benchmarks.

2. Related Work

Many methods extract, learn, or reason over entire body features. Some use a combination of local detectors and structural constraints [23] for coarse tracking or for person dependent tracking [4]. Methods using "Pictorial Structures", such as [12], made this approach tractable with so called "Deformable Part Models (DPM)". Subsequently a large number of related models were developed [8, 30, 10]. Algorithms which model more complex joint relationships, such as [30], use a flexible mixture of templates modeled by linear SVMs. A cascade of body part detectors to obtain more discriminative templates was employed in [16]. Most recent approaches aim to model higher-order part relationships. A model that augments the DPM model with Poselet [3] priors was proposed in [19, 20] in order to capture spatial relationships of body-parts. A multi-modal model which includes both holistic and local cues for mode selection and pose estimation was proposed in [24]. Following the Poselets approach, the Armlets approach in [13] employs a semi-global classifier for part configuration and shows good performance on real-world data. This approach exhibits good performance on real-world data, however it is demonstrated only on arms. All these approaches use hand crafted features (i.e. edges, contours, HoG features and color histograms), which have been shown to have poor generalization performance and discriminative power.

With the introduction of "DeepPose" in [27], the research on human pose estimation shifted to deep network approaches. A network to directly regress the 2D coordi-

nates of joints was used in [27]. In addition to the use of graphical models, there are several examples of iterative or multi-stage training methods in a sequential, cascaded fashion [28].

In [6], the ConvNet predictions were improving iteratively in a process called Iterative Error Feedback (IEF). Each successive run through their network takes as input the image along with predictions from the previous forward pass and further refines them. This way it iteratively improves part detections using error feedback, but uses a Cartesian representation as in [27] which does not preserve spatial uncertainty and results in lower accuracy in the high precision regime. In [28], an extension based on the work of multi-stage pose machines [22] by using ConvNets for feature extraction without an explicit graphical model-style inference was proposed. A "stacked hourglass" network design for predicting human pose was proposed in [17]. This network tries to capture and consolidate information across all scales of the image by pooling down to a very low resolution, then upsampling and combining features across multiple resolutions.

The combination of a low-dimensional representation of the input image produced by a ConvNet in [7] and an image dependent spatial model show improvement over the work proposed in [27]. In other words, detections were clustered into typical orientations so that when their classifier makes predictions additional information is available indicating the likely location of a neighbouring joint. In the literature, multi-resolution ConvNet architectures were developed in order to perform heat-map likelihood regression for each joint (rougher pose estimators). These architectures were trained jointly with a MRF-based spatial model network [26] or with a pose refinement model [25]. Others have recently tackled the problem of learning typical spatial relationships between joints in similar ways [11, 21] with variations on how to approach unary score generation and pairwise comparison of adjacent joints. Similarly, motion features can be added to the input of a multi-resolution ConvNet architecture to further improve accuracy [14]. In [5], a ConvNet cascaded architecture designed for learning part relationships and spatial context is presented.

3. Model Architecture

The overall architecture is shown in Figure 2. It consists of a coarse heat-map regression model, our proposed spatial geometric model, the module to sample and crop the convolutional feature maps at a specified (x, y) location for each joint, and the fine heat-map regression (coarse to fine) model. In this Section we give a description of each ConvNet used in our framework focusing on the proposed part-based spatial model.

3.1. Coarse Heat-Map Regression Model

The coarse heat-map regression model takes as input an RGB Gaussian pyramid of three levels (in Figure 3 only two levels are shown for brevity) and for each body joint it outputs a heat-map, that is a per-pixel likelihood that the joint in question is depicted at that location. We use an input resolution of 256×256 pixels at the highest level of the pyramid. The first layer of the network performs local contrast normalization (LCN) using the same filter kernel in each of the three resolution banks. Each LCN image is then input to a ten layer multi-resolution ConvNet. Due to the presence of pooling the output heat-map is at a lower resolution than the input image.

3.2. Part-based Spatial Model

In this Section we describe in detail the spatial model that introduces the geometric constraints between the body parts. Our model, depicted in Figure 4, builds on the MRF-based spatial model proposed in [25, 26], that formulates a tree-structured MRF over spatial locations using a random variable for each joint. In that formulation, the message passing that performs inference is expressed using convolutional filtering operations and therefore can be implemented as a specialized layer in a ConvNet. In this way the filters that produce the unary and the pairwise potentials of the MRF model can be learned by supervised training, either of the last layer, or of the whole network in an end-to-end fashion. For our 32×32 pixel heat-map input to this model, this results in large 63×63 convolution kernels to account for a joint displacement radius of maximum 32 pixels. The convolution sizes are adjusted so that the largest joint displacement is covered within the convolution window. In such a network, the filters, denoted by $f_{a|c}$, are functions of the conditional probabilities $e_{a|c}$ of the location of joint a, given the location of another joint c. That is, the refinement $\bar{e}_a$ of the heat-map for a joint a, is given by filtering operations on functions of the heatmaps e_c of the other joints c. More specifically,

$$\bar{e}_a = exp\Big(\sum_{c \in V} \log[f_{a|c} * ReLU(e_c) + SoftPlus(b_{a|c})]\Big), \tag{1}$$

where $f_{a|c} = SoftPlus(e_{a|c})$ (see [26] for more details).

A major drawback with such an approach is that a single filter, i.e. $f_{a|c}$ is used to model the pairwise relations between joints. In the case of a dataset containing a large variety of poses (e.g. both standing and laying) this results with rather uninformative filters. To deal with this problem the proposed MRF-based loopy belief propagation network is constrained by a low dimensional latent model. In the proposed model, each of the filters $f_{a|c}$ is a linear combination of K filters $f^k_{a|c}$, where the weights $\mathbf{w} \in R^K$ of this linear combination are determined by the projection of

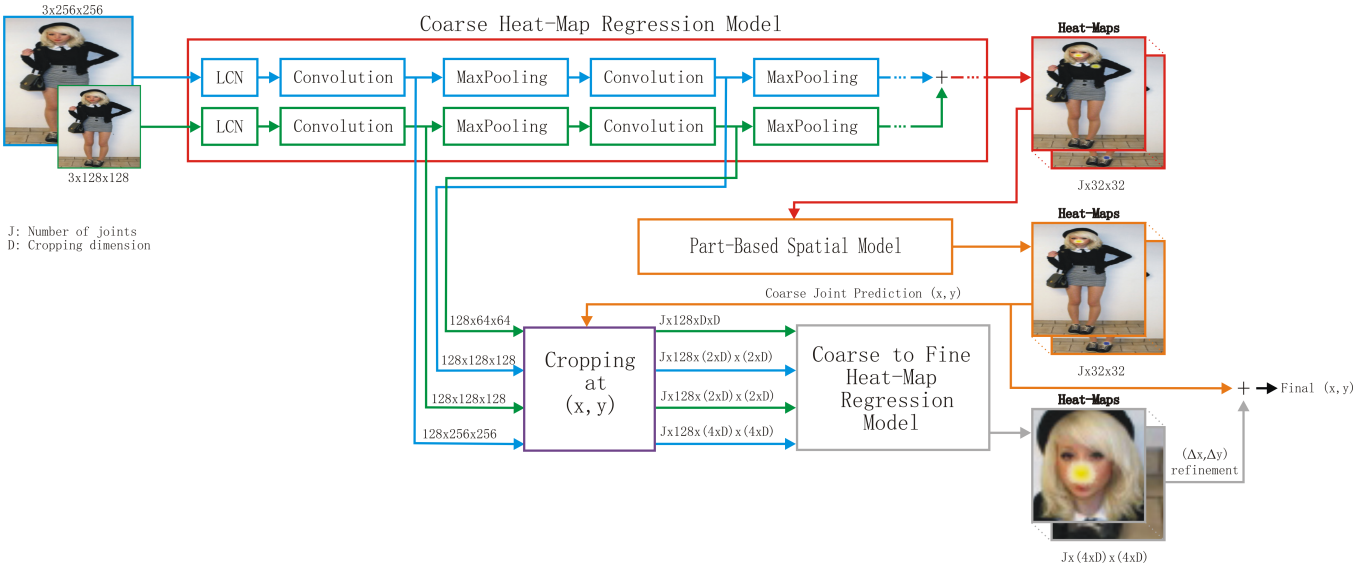


Figure 2. Overview of our unified learning framework.

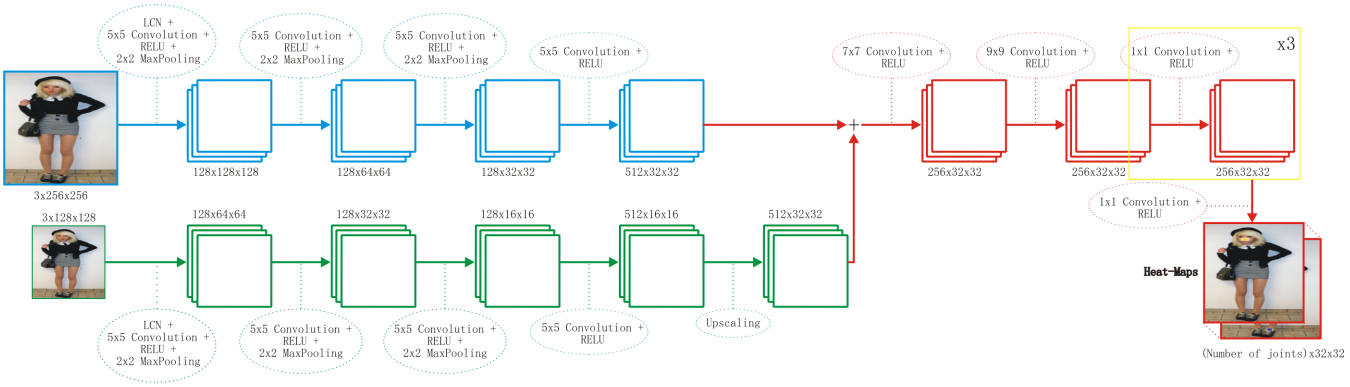


Figure 3. Architecture of our coarse heat-map regression model.

the heat-maps into a K dimensional manifold that encodes global constraints, such as the global pose. That is:

$$f_{a|c} = \sum_{k=1}^{K} \mathbf{w}_k * f_{a|c}^k = \mathbf{w}^T \begin{bmatrix} f_{a|c}^1 \\ . \\ . \\ . \\ f_{a|c}^K \end{bmatrix}. \quad (2)$$

The projection of the heat-map volume to the low dimensional manifold, that is the calculation of the weights $\mathbf{w}$, is performed by a separate branch of the network that performs dimensionality reduction on the heat maps. It consists of convolutional and fully-connected layers and is depicted as the lower branch in Figure 4. The parameters of that branch are jointly trained with the main network using both a discriminative and a generative cost - the latter being essentially a classical auto-encoder cost. Thus, the weights $\mathbf{w}$ are learned by a cost function that combines both a generative term that comes from the auto-encoder ConvNet and a discriminative cost that comes from the heat-map prediction. In this way, the conditionals become more informative as they attempt to model pairwise relations under specific global constraints as those are encoded in the coordinates $\mathbf{w}$ at the global pose manifold.

Another drawback of the baseline model of Eq. 1 is that it assumes that the learned pair-wise joint distributions/relations should contribute equally to marginal likelihood of location of a joint. We relax this assumption by applying, for each of the K dimensions of the pose manifold, a weighting scheme that determines the strength of the joints' spatial relationships. That is, we allow that, conditioned on a global pose, some pairwise relations between different joints are more informative that others. This is expressed as a filtering operation with weights $\beta_{a|c}^k$. That is:

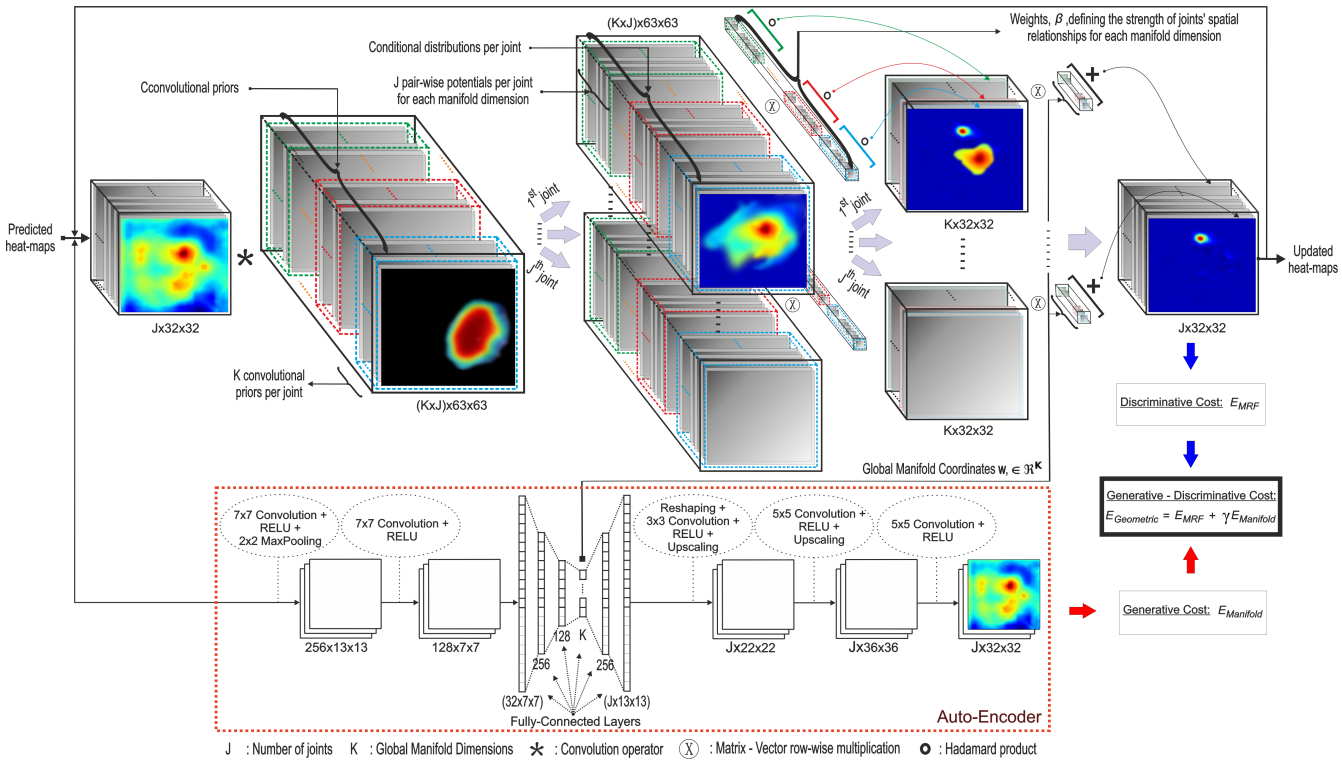


Figure 4. The proposed constrained convolutional MRF-part based spatial model architecture. The lower branch is an auto-encoder ConvNet which learns the low K^{th} dimensional pose manifold. The weights $\mathbf{w} \in R^K$ are learned by a cost function that combines both a generative term that comes from the auto-encoder ConvNet and a discriminative cost that comes from the heat-map prediction.

$$\bar{e}_a = exp\Bigg(\sum_{c \in V} \Bigg[\sum_{k=1}^{K} \beta_{a|c}^k \log \Big[\mathbf{w}_k * f_{a|c}^k * ReLU(e_c) + SoftPlus(b_{a|c}) \Big] \Bigg] \Bigg), \quad (3)$$

The weights $\beta_{a|c}^k, (1 \leq k \leq K)$ are learned jointly with the other parameters of the network using back-propagation. Note, that $\mathbf{w}$ are not fixed weights that are learned during training and fixed during testing, but weights that are estimated at test time, by the auto-encoder ConvNet.

Finally, the baseline model of [26] applies only one step of the MRF-based inference. Inspired by the ConvNet in [6] that uses a self-correcting model that progressively changes an initial solution by feeding back error predictions, we apply the filtering steps of Eq. 3 in an iterative manner updating the same $f_{a|c}$, $b_{a|c}$ and $\beta_{a|c}$ parameters. That is, the output heat-maps of the proposed MRF-part based spatial model are progressively changing by being fed back to the model as inputs. This is depicted by the feedback loop in Figure 4.

3.3. Fine Heat-Map Regression Model

The goal of using a fine regression model is to recover the spatial accuracy lost by pooling in the coarse regression model. Thus, an additional ConvNet proposed in [25] was used to refine the localization result of the unified coarse model. More specifically, by reusing existing convolution features this model is trained to estimate the joint offset location within a small region of the image extracted around the estimates of the unified coarse model, reducing in that way the number of trainable parameters in the cascade. This network outputs a high resolution per-pixel heat-map which corresponds to this small region, that is a per-pixel likelihood for key joint locations on the human skeleton.

4. Training and Data Augmentation

4.1. Model Training

All of the ConvNets described above do not estimate the positions of the body joints directly [18, 27], but estimate instead one heat-map for each of the joint positions. Those heat-maps (i.e. the output of last convolutional layer) form a fixed-size $M \times N \times J-$dimensional tensor (here $32 \times 32 \times J$), where M, N and J denote the height, the width and the number of joints, respectively. In case of the coarse heat-map regression model and the MRF-part based spatial model the output heat-maps have fixed spatial dimensions, M=N=32, while in case of the fine heat-map regression model these two dimensions depend on the size of the cropping region as described before.

At training time, the ground truth labels for all ConvNets are heat-maps that are constructed for each joint separately by placing a Gaussian with fixed variance ($\sigma \approx 1.5$ pixels) at the ground truth position of the corresponding joint. We then use an ℓ_2 loss, that is we optimize the sum of the squared pixel-wise differences between the output heat-map and the constructed ground truth heat-map.

Let us denote by (I_i, C_i) the i-th training example, where $C_i \in R^{2J}$ denote the coordinates of the J joints in the image I_i. Given a training dataset $N = \{(I_i, C_i)\}$ and the ConvNet regressor ϕ (the output of last convolutional layer), we train our ConvNet by estimating the network weights p that minimize the objective function $E_{<ConvNet>}$:

$$E_{<ConvNet>} = \sum_{(I,C)\in N} \sum_{m,j} \|H_{m,n,j}(C_j) - \phi_{m,n,j}(I,p)\|^2, \tag{4}$$

where $H_{m,n,j}(C_j) = \frac{1}{2\pi\sigma} e^{-[(C_j^1 - m)^2 + (C_j^2 - n)^2]/2\sigma}$ is a Gaussian centred at C_j with σ fixed. Then, E_{Coarse}, $E_{Geometric} = E_{MRF} + \gamma E_{Manifold}$ and $E_{Coarse2Fine}$ denote the objective function for each of our three ConvNets. $E_{Manifold}$ denotes the objective function for the auto-encoder ConvNet which creates the low dimensional pose manifold, while γ is a constant used to provide a trade-off between the relative importance of the two sub-tasks.

4.2. Joint Inference And Training

Given an input image, the joint inference is done as follows. First we do forward propagation through the coarse heat-map model and our geometric model and infer all joint (x, y) locations by finding the maximal value in each joint's heat-map. This coarse (x, y) location is then used to sample and crop the first two convolutional layer feature maps at each of the joint locations. We do this for all the resolution banks, keeping the contextual size of the window constant by scaling the cropped area at each higher resolution level. After that, the resulting features are further propagated through a fine heat-map model to give a (Δx, Δy) offset within the cropped sub-window. Finally, by adding the position refinement to the coarse location we end up with the final (x, y) location prediction for each joint.

Regarding the joint training, our proposed constrained convolutional MRF-part based spatial network is combined with the coarse heat-map regression model described in section 3.1 into a single unified coarse heat-map regression model. This is done by firstly training the coarse heat-map regression model separately by minimizing E_{Coarse} and storing the heat-map outputs. The outputs are then used to train firstly our pose manifold generator by minimizing $E_{Manifold}$, and secondly, our geometric model by minimizing $E_{Geometric}$ (we used γ = 0.4). After that, the trained coarse and geometric model are combined and fine-tuned using back-propagation through the unified coarse heat-map regression model by minimizing $E_{Unified\_Coarse} = E_{Coarse} + E_{Geometric}$ and storing the heat-map outputs. Subsequently, the outputs are used to train the coarse-to-fine heat-map regression model by minimizing $E_{Coarse2Fine}$. After that, the trained unified coarse and coarse-to-fine models are combined and jointly fine-tuned using back-propagation through the unified coarse heat-map regression model by minimizing $E_{Unified} = E_{Unified\_Coarse} + \lambda E_{Coarse2Fine}$, where λ is a constant used to provide a trade-off between the relative importance of the two sub-tasks. λ is another network hyper-parameter and is chosen to optimize performance over the validation set (we used λ = 0.25). This unified fine-tuning further improves performance, because the geometric model is able to effectively reduce the output dimension of possible heat-map activations and therefore the coarse model can use the available learning capacity to better localize the precise target activation.

Table 1. Window sizes that were used for the different body joints at the higher resolution input image.

Cropping Window Size (in pixels) Per Joint						
Head	**Shoulder**	**Elbow**	**Hip**	**Knee**	**Wrist**	**Ankle**
27	27	36	36	45	54	63

In practice, many of the failure cases were caused by either an occluded or a mis-attributed limb and refinement of the position within a local window would not result in improvements. In both cases the prediction error was large and therefore the small fixed window used in [25], would not include the correct target location and the refinement model could not therefore lead to an improved estimation. For this reason, in [25] the contribution of this part of the network architecture is small (λ = 0.1). In this work, we do not use windows of fixed length to ensure that in the vast majority of cases (more than 95% in the training set), the true target location is within the used window. This way, we overcome the problems that [25] faces in the case of occlusions and in the case that the coarse model provides estimates that are far from the true target location, and rely more (λ = 0.25) on the refinement model when training the proposed architecture in an end-to-end fashion. In Table 1, we report the window sizes that were used for different body joints at the higher resolution input image.

In order to better exploit the fine heat-map model by keeping at the same time the cropping regions small we used the training procedure described below. In the beginning of the training procedure, only the images with small prediction error were used. Once the joint estimation accuracy on the training data was significantly improved by the ConvNet, the rest of the images were gradually included, based on the corresponding prediction errors. This is also important since in the beginning we used quite a large learning rate, while when the most difficult

images were processed the learning rate was significantly decreased. During each training/validation iteration, each input image is randomly rotated (with $r \in [-30^o, +30^o]$), scaled (with $s \in [0.8, 1.2]$), sheared (with shear factor in pixels $\in [-3, 3]$), stretched (with stretching factor equal to 1.2^{-1}) and flipped horizontally (with probability equal to 0.5) - those transformations are introduced in order to improve the generalization performance on the validation set. In addition, for images whose prediction error was significantly higher than the mean data prediction error, we apply more than one random image transformation (two in our experiments). Finally, we use more than one random image transformation per image in the validation procedure too - we used four in our experiments.

5. Evaluation

5.1. Datasets / Training Details

In this work we used the FashionPose [8], the MPII [1] and the LSP [16] databases. FashionPose dataset consists of 7,543 accurately annotated images downloaded from a variety of fashion blogs and it is annotated by 13 joints. MPII Human Pose dataset is the most diverse set of human pose-labeled images, it is a full-body dataset and it is a video dataset. This dataset is very challenging and it includes a wide variety of full-body pose annotations within the 28,821 training and 11,701 test examples. LSP dataset consists of 11,000 images for training and 1,000 images for testing and is annotated by 14 joints.

We implemented our network using the Lasagne library within the Theano [2] framework and optimized the parameters using Adagrad [9]. The training of the coarse heat-map regression model takes approximately 4 days, the part-based spatial model 3 days and the coarse to fine heat-map regression model takes 4 days on a 12GB Nvidia Tesla K80 GPU. The forward-propagation for a single image through all networks takes around 125ms. For MPII, it is standard to utilize the scale and center annotations provided with all images. All images were cropped after centering on the person and then scaled to get a 256×256 input for the network such that a standing-up human has height 200 pixels. In case of severely occluded joints we used a ground truth heat-map of all zeros for supervision.

5.2. Experimental Results

In order to qualitatively show the complexity of the used datasets and illustrate the performance of our method, in Figure 5 we depict some examples where our system estimates the human pose well. For generating final test predictions we run both the original input and a flipped version of the image through the network and average the heat-maps together [29]. The chosen examples have PCK-0.5 error less than 0.15, that is, the average error for all joints is less than 0.15 of the half body height.

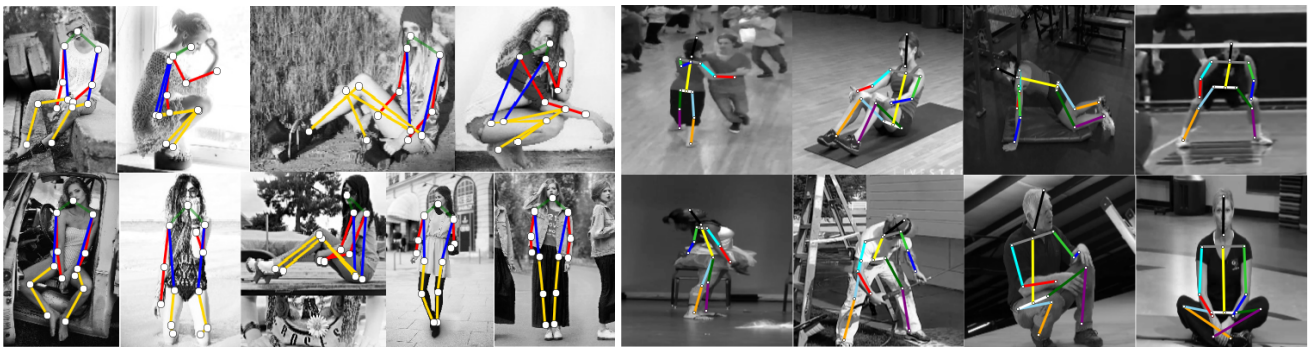
In order to show the influence of our contributions and compare our results with [25] and [26], we report the PCK (Probability of Correct Keypoints [8]). In Table 2 we summarize the results at accuracy of PCK = 0.15. In order to show the influence of the individual contributions, we report results for the MPII database for a) the coarse model (CM), b) the coarse plus the coarse to fine models (CM + C2FM), c) the full model comprising of the coarse plus the MRF plus the coarse to fine models (CM+MRF+C2FM) (full model) with only one iteration of our MRF model, d) the full model when one filter is used to model the joint pairwise potentials (K=1), e) the full model when K=4, f) the coarse model and g) the full model of [25]. It is clear that in both datasets our coarse model outperforms the coarse model of [25], illustrating the influence of the proposed architectural changes in the size of depth and filter size of the coarse ConvNet. The results also show the influence of the proposed contributions after the coarse model since both the iterative MRF process as well as the constrained MRF model significantly improve the performance of the system. Our full model improves our coarse model 2.46% more than the full model of [25] over its coarse model. Note that, as described in Section 3.2, even when K=1 our MRF model is roughly equivalent to [25]. To limit the framework's complexity we did not perform experiments for K >4. Considering that (a) the coarse to fine model is a siamese ConvNet, and (b) in the iterative process of our geometric model we use weight sharing, the total number of our training parameters is similar to other state-of-the-art techniques.

Table 2. Comparison with prior-art. PCK @ 0.15 for MPII and FashionPose Database compared to the state-of-the-art methods

MPII Database	
Methods	**Full Body**
Tompson et al., CVPR 2015 - CM ConvNet	36.25
Tompson et al., CVPR 2015 - Full Model	44.08
Andriluka et al., CVPR 2009	14.94
Toshev et al., CVPR 2014	24.80
Our System - CM ConvNet	38.71
Our System - (CM+C2FM) ConvNet	45.71
Our System - (CM+C2FM+MRF) ConvNet	47.49
Our System - Full Model (one MRF loop)	48.40
Our System - Full Model when K=1	48.60
Our System - Full Model when K=4	**49.62**
Our System - Full Model with data augmentation of Tompson et al., CVPR 2015	48.98

FashionPose Database	
Methods	**Full Body**
Dantone et al., PAMI 2014	63.92
Our System - CM ConvNet	84.55
Our System - (CM+C2FM) ConvNet	86.95
Our System - Full Model	**90.21**

The proposed architecture introduces the ConvNet with geometric constraints before the refinement ConvNet. This is in contrast to other methods in the literature, e.g. [25] that introduce such constraints at the final layers of their



(a) (b)

Figure 5. Human pose estimation (PCK-0.5 error<0.15) on sample images from (a) FashionPose and (b) MPII testing datasets.

Table 3. Comparison with prior-art. Error per joint for the MPII dataset compared to the state-of-the-art methods.

	PCKh @ 0.15								PCKh @ 0.5							
Methods	**Head**	**Shoulder**	**Elbow**	**Wrist**	**Hip**	**Knee**	**Ankle**	**Full Body**	**Head**	**Shoulder**	**Elbow**	**Wrist**	**Hip**	**Knee**	**Ankle**	**Full Body**
Wei et al., CVPR 2016	64.6	62.1	55.8	50.5	55.0	49.7	46.5	55.4	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.9
Pishchulin et al., CVPR 2016	61.2	57.4	50.6	43.9	49.3	42.9	35.3	49.4	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Carreira et al., CVPR 2016	62.3	58.8	48.4	39.6	49.9	40.2	33.5	48.3	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Bulat et al., ECCV 2016	64.7	61.8	56.9	52.3	56.0	52.3	49.1	55.4	97.8	95.1	89.9	85.3	89.4	85.7	81.9	89.6
Newell et al., ECCV 2016	65.2	62.9	58.4	53.8	56.9	53.8	50.6	57.3	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Tompson et al., NIPS 2014	63.1	57.5	47.2	41.7	44.8	36.5	30.1	46.8	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6
Tompson et al., CVPR 2015	63.3	58.4	51.1	45.1	47.3	38.9	31.3	48.9	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Our System - Full Model	**65.8**	**63.7**	**58.9**	**54.7**	**57.9**	**54.4**	**51.9**	**58.21**	**99.1**	**97.2**	**93.3**	**88.9**	**91.9**	**88.4**	**86.1**	**92.1**

Table 4. Comparison with prior-art. Error per joint for the LSP dataset compared to the state-of-the-art methods.

	PCK@0.2							
Methods	**Head**	**Shoulder**	**Elbow**	**Wrist**	**Hip**	**Knee**	**Ankle**	**Full Body**
Wei et al., CVPR 2016	97.8	92.5	87.0	83.9	91.5	90.8	89.9	90.5
Pishchulin et al., CVPR 2016	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Carreira et al., CVPR 2016	90.5	81.8	65.8	59.8	81.6	70.6	62.0	73.1
Bulat et al., ECCV 2016	96.3	92.2	88.2	85.2	92.2	91.5	88.6	90.7
Tompson et al., NIPS 2014	90.6	79.2	67.9	63.4	69.5	71.0	64.2	72.3
Our System - Full Model	**97.9**	**93.6**	**90.1**	**87.1**	**94.2**	**93.2**	**90.5**	**92.4**

networks. The motivation for doing so is that in our architecture the main purpose of that ConvNet is to provide a better initial estimate such that the refinement network can provide an accurate estimate within a window that contains high resolution image information. In order to validate our choice, we provide results in Table 2 when a ConvNet that introduces geometric constraints is placed after the refinement ConvNet (CM+C2FM+MRF). Our choice is justified by the fact that the performance of the overall system drops considerably from 49.62% to 47.49% when the geometric constraints are introduced at the final layers of our networks. In order to validate that our data augmentation procedure enhances the performance of our model, in Table 2 we provide the performance of our model when the augmentation procedure of [25] is used. In this case, the performance of the overall system drops from 49.62% to 48.98%

In Tables 3 and 4 we report the error per joint for the MPII and LSP datasets - as reported in other works in the literature, wrists and angles that exhibit larger variations in their motion are the ones that are harder to localize. Based on the experimental results, it is clear that the proposed unified learning framework outperforms existing state-of-the-art techniques on both of these challenging datasets. Furthermore, the performance of our system is considerably better in the case of the harder joints (i.e. arms, wrists and ankles) even at high levels of accuracy.

6. Conclusions

In this paper, we presented a cascaded architecture for human body pose estimation that combines coarse, part-based spatial models and fine scale ConvNets. This work introduces a MRF-based spatial ConvNet between the coarse and the refinement model that introduces geometric constraints. We propose an MRF architecture in which a) the filters that implement the message passing in the MRF inference are factored so as to be constrained by a low dimensional pose manifold the projection to which is estimated by a separate branch of the proposed ConvNet, and b) the strength of the pairwise joint constraints are modeled by weights that are jointly estimated with the other parameters of the network. These three ConvNets were trained into a unified learning framework achieving state-of-the-art results on challenging datasets for human pose estimation.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and S. Bernt. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1, 7

[2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010. 7

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. *IEEE International Conference on Computer Vision (ICCV)*, 2009. 2

[4] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching tv (using weakly aligned subtitles). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[5] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 3

[6] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 5

[7] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014. 1, 3

[8] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Body parts dependent joint regressors for human pose estimation in still images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014. 2, 7

[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 12:2121–2159, 2011. 7

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html. 2

[11] X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. *arXiv preprint arXiv:1504.07159*, 2015. 3

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[13] G. Gkioxari, P. Arbeláez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[14] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. *arXiv preprint arXiv:1409.7963*, 2014. 1, 3

[15] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010. 1

[16] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1465–1472, 2011. 2, 7

[17] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[18] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Asian Conference on Computer Vision (ACCV)*, 2014. 5

[19] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[20] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3487–3494, 2013. 2

[21] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[22] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision (ECCV)*, 2014. 3

[23] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

[24] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2

[25] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 3, 5, 6, 7, 8

[26] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Join training of a convolutional network and a graphical model for human pose estimation. In *Neural Information Processing Systems (NIPS)*, 2014. 1, 2, 3, 5, 7

[27] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2, 3, 5

[28] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[29] H. Yang and I. Patras. Mirror, mirror on the wall, tell me, is the error small? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4693, 2015. 7

[30] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(12):2878–2890, 2013. 2